

Claims

[c1] 1. A method for creating an application program, the method implemented on a computer system having persistent storage, a display screen and one or more input devices, the input devices controllable by a user to create visual representations on the display screen, the method comprising

- A. defining and supporting a set of pre-developed object classes, the said pre-developed object classes are all derived from one generic class which supports a property-method-event model; defining an action class and an action list class; the action class has, as its members, action performer, action method, and action data; the action list class contains a sorted list of action class instances; the action performer is one of the pre-developed object class; the action method is one of the methods supported by the action performer; the action data are the parameters needed by the action method;
- B. generating and graphically displaying, in response to input from the user, instances of the object classes from the said pre-developed object classes;
- C. setting, in response to input from the user, each property of each instance of the object classes created in step B;

2. The method of claim 1 further comprising the step of:

- D. creating, in response to input from the user, instances of the action list class which contains a sorted list of instances of the action class;

Wherein step D comprises

- D1. Creating, in response to input from the user, each action class instance of each action list class instance; and

Wherein step D1 comprises:

- D1a. Selecting an instance of object class from the instances of object classes created in step B; the said selected instance of object class is used as the action performer member for the action class instance;
- D1b. selecting a method from the methods supported by the instance of object class selected in step D1a; the

selected method is used as the action method for the action class instance;

D1c. according to the method selected in step D1b, it is known the number and types of the parameters needed for the said method; if one or more parameters are needed for the said method, then one or more dialog-boxes are provided for the user to specify the appropriate parameters for the method;

3. The method of claim 1 further comprising the step of

E. linking, in response to input from the user, action list instances created in step D to events of the instances of the object classes to form an event-action-list mapping;

Wherein step E comprises:

E1. Selecting, in response to input from the user, an instance of object class from the existing instances of object classes;

E2. Selecting, in response to input from the user, an event from the events supported by the object class instance selected in step E1;

E3. Selecting, in response to input from the user, an action list class instance from the action list class instances created in step 2;

E4. Building the mapping relationship between the action-list selected in step E3 and the event selected in step E2;

4. The method of claim 1 further comprising the step of:

F. Selecting, in response to input from the user, a set of object class instances to be specified as the "initially active object class instances" usually the object class instances presented on the first application screen presentation is such a set of the "initially active object class instances"

5. The method of claim 1 further comprising the step

G. saving to the computer persistent storage the object class instances

created in steps A, B and C, the action list class instances created in step D, the mapping relationship built in step E between the events of object class instances and the action lists, indication of which object instances are the "initially active object class instances" as specified in step F;

6. The method of claim 1 further comprising the step of:

H. an execution environment;

Wherein step H comprises:

H1. Reading back from the computer persistent storage the object class instances created in steps A, B and C, the action lists created in step D, the mapping relationship built in step E between the events of the object class instances and the action list, indication of which object instances are the "initially active object class instances" as specified in step F;

H2. Creating and displaying the said "initially active object class instances"

H3. Responding to each event fired by each object class instance;

Wherein step H3 comprises:

H3a. Checking if there is a mapping relationship between an action list class instance and the said event;

H3b. If the said mapping relationship exists, sequentially performing each action in the said action list mapped to the said event;

H3c. Each action in the said action list is performed by the following steps:

H3c1. Locating the object class instance which is assigned as the action performer for the action;

H3c2. Signaling to the said action performer which action method is specified for the action;

H3c3. If there are method data specified for the said method of the said located object class instance, the

method data are passed to the said object class instance as well;

H3C4. The said located object class instance carries out the said action method.

7. The method of claim 1 further comprising the step of:

- I. A context-data buffer which saves event parameter data such as mouse position in mouse movement events; every time an event is fired, before an action list is executed as an event handler, the said context-data buffer is filled with the said event parameter data;
- J. The said context-data buffer is available for the user to pick as the method data in step D1c;

[c2] 8. A method designed for object classes to dynamically change their event-action-list mapping at the runtime; any object classes may choose to support or not to support the said method; the said method has two parameters; the first parameter is the event identifier which identifies an event supported by the object class which is the owner of the said method; the second parameter is the action list class instance identifier which identifies an action list class instance;

9. The method of above claim further comprising the step of:

- K. At the runtime, when an object class instance is asked to perform the said method , the said object class instance uses the first parameter of the said method to locate its event;
- L. The said object class instance uses the second parameter of the said method to locate the action list;
- M. The said object class instance rebuilds the event-action-list map using its event located in step K and the action list located in step L.